

BEST AVAILABLE COPY

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 10214217 A

(43) Date of publication of application: 11.08.98

(51) Int. Cl

G06F 12/00

G06F 12/00

G06F 13/00

(21) Application number: 09015644

(71) Applicant: SHARP CORP

(22) Date of filing: 29.01.97

(72) Inventor: MIZOTE KOICHI

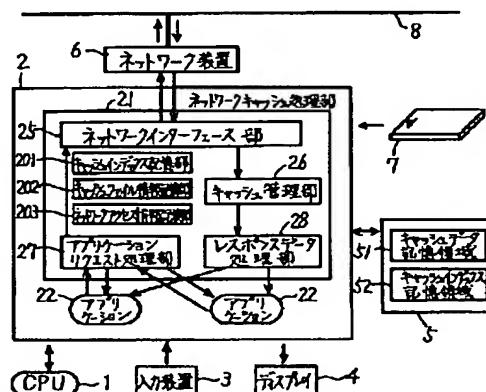
(54) **NETWORK CACHE DEVICE AND METHOD
THEREFOR**

COPYRIGHT: (C)1998,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a network cache device in which plural applications can share the same cache data, and each application can quickly receive response data.

SOLUTION: This device includes a network interface part 25 which issues a request for data access to a network 8, and receives response data corresponding to the request, outside storage device 5 which stores the response data received from the network 8, and network cache processing part 21 which receives a data request by inter-process communication from an application 22, does not instruct the issue of the request for the data access to the network interface part 25 when the response data corresponding to the data request are stored in the outside storage device 5, and transmits the response data corresponding to the data request stored in the outside storage device 5 to the application 22.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-214217

(43)公開日 平成10年(1998)8月11日

(51)Int.Cl.⁶
G 0 6 F 12/00
13/00

識別記号
5 4 5
5 1 4
3 5 4

F I
G 0 6 F 12/00
13/00

5 4 5 B
5 1 4 K
3 5 4 D

審査請求 未請求 請求項の数10 O.L (全 19 頁)

(21)出願番号 特願平9-15644

(22)出願日 平成9年(1997)1月29日

(71)出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72)発明者 溝手 弘一

大阪府大阪市阿倍野区長池町22番22号 シ
ャープ株式会社内

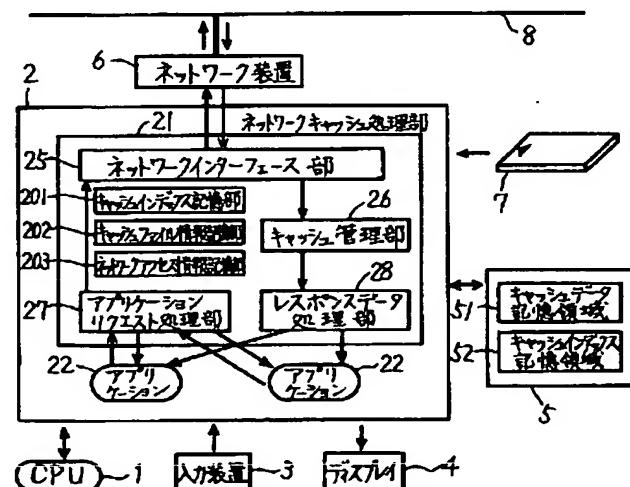
(74)代理人 弁理士 深見 久郎

(54)【発明の名称】 ネットワークキャッシュ装置およびネットワークキャッシュ方法

(57)【要約】

【課題】 複数のアプリケーションが同じキャッシュデータを共有可能であり、各アプリケーションがレスポンスデータを速く受信することが可能なネットワークキャッシュ装置を提供すること。

【解決手段】 ネットワーク8に対してデータアクセスのリクエストを発行し、リクエストに対するレスポンスデータを受信するためのネットワークインターフェース部25と、ネットワーク8から受信したレスポンスデータを記憶するための外部記憶装置5と、アプリケーション22からプロセス間通信によりデータリクエストを受信し、データリクエストに対応するレスポンスデータが外部記憶装置5に記憶されている場合は、ネットワークインターフェース部25に対してデータアクセスのリクエストの発行を指示せず、外部記憶装置5に記憶された当該データリクエストに対応するレスポンスデータをアプリケーション22へ送出するためのネットワークキャッシュ処理部21とを含む。



【特許請求の範囲】

【請求項1】 ネットワークに対してデータアクセスのリクエストを発行し、該リクエストに対するレスポンスデータを受信するためのネットワークインタフェース手段と、

前記ネットワークインタフェース手段が受信したレスポンスデータを記憶するための記憶手段と、

アプリケーションからプロセス間通信によりデータリクエストを受信し、

前記データリクエストに対応するレスポンスデータが前記記憶手段に記憶されている場合は、前記ネットワークインタフェース手段に対してデータアクセスのリクエストの発行を指示せず、

前記データリクエストに対応するレスポンスデータが前記記憶手段に記憶されていない場合は、前記ネットワークインタフェース手段に対してデータアクセスのリクエストの発行を指示するためのキャッシング処理手段とを含むネットワークキャッシング装置。

【請求項2】 前記キャッシング処理手段は、前記リクエストに対応するリクエストデータ名と前記リクエストに対応するレスポンスデータが前記記憶手段内に記憶される領域とがインデックス情報として記憶されるインデックス記憶手段と、

前記ネットワークインタフェース手段によって受信されたレスポンスデータを前記記憶手段に記憶し、該記憶されたレスポンスデータに対応するインデックス情報を前記インデックス記憶手段に記憶するためのキャッシング管理手段と、

前記アプリケーションからプロセス間通信によりデータリクエストを受信し、

前記データリクエストに対応するリクエストデータ名が前記インデックス記憶手段にある場合は、前記記憶手段に記憶された前記データリクエストに対応するリクエストデータをプロセス間通信により前記アプリケーションへ送出し、

前記データリクエストに対応するリクエストデータ名が前記インデックス記憶手段にない場合は、前記ネットワークインタフェース手段に対してデータアクセスのリクエストの発行を指示するためのアプリケーションリクエスト処理手段と、

前記ネットワークインタフェース手段によって受信されたレスポンスデータをプロセス間通信により前記アプリケーションへ送出するためのレスポンスデータ処理手段とを含む、請求項1記載のネットワークキャッシング装置。

【請求項3】 前記レスポンスデータ処理手段は、前記ネットワークインタフェース手段によって受信されたレスポンスデータを共有メモリ領域にコピーし、前記レスポンスデータがコピーされた領域へのポインタと前記レスポンスデータのデータサイズとを前記アプリケーショ

ンへメッセージ通信する、請求項2記載のネットワークキャッシング装置。

【請求項4】 前記キャッシング処理手段はさらに、前記ネットワークインタフェース手段が前記ネットワークに対してデータアクセス中のリクエストの情報が記憶されるネットワークアクセス情報記憶手段を含み、

前記アプリケーションリクエスト処理手段は、前記アプリケーションからプロセス間通信によりデータリクエストを受信し、

10 前記データリクエストに対応するリクエストデータ名が前記インデックス記憶手段にある場合は、前記記憶手段に記憶された前記データリクエストに対応するリクエストデータをプロセス間通信により前記アプリケーションへ送出し、

前記データリクエストに対応するリクエストデータ名が前記インデックス記憶手段になく、前記ネットワークアクセス情報記憶手段にもない場合は、前記ネットワークインタフェース手段に対してデータアクセスのリクエストの発行を指示し、前記ネットワークアクセス情報記憶

20 手段に当該データリクエストの情報を記憶し、

前記データリクエストに対応するリクエストデータ名が前記インデックス記憶手段になく、前記ネットワークアクセス情報記憶手段にある場合は、前記ネットワークアクセス情報記憶手段に記憶されている当該リクエストのデータアクセスが終了するまで前記データリクエストに対する処理を延期する、請求項2記載のネットワークキャッシング装置。

【請求項5】 前記キャッシング管理手段は、前記インデックス記憶手段にインデックス情報を記憶する際、前記記憶手段にも同じインデックス情報を記憶し、前記記憶手段に格納されたリクエストデータを更新する際更新された情報を附加したインデックス情報を前記記憶手段に追加し、所定のタイミングで前記更新された情報を附加したインデックス情報を参照して前記記憶手段に記憶されたインデックス情報を再構築する、請求項2記載のネットワークキャッシング装置。

【請求項6】 前記キャッシング管理手段は、前記インデックス記憶手段にインデックス情報を記憶する際当該リクエストデータがアクセスされた時刻も記憶し、前記記憶手段に記憶されたリクエストデータの総容量が所定値を超えた場合に前記総容量が前記所定値以下になるまで前記アクセスされた時刻が古いリクエストデータを順次消去する、請求項2記載のネットワークキャッシング装置。

【請求項7】 前記キャッシング管理手段は、前記ネットワークインタフェース手段によって受信されたレスポンスデータを前記記憶手段に記憶する際、1つのキャッシングファイルに複数のレスポンスデータを記憶する、請求項2記載のネットワークキャッシング装置。

50 【請求項8】 前記キャッシング処理手段はさらに、前記

キャッシュファイルがアクセスされた時刻が前記キャッシュ管理手段によって記憶されるキャッシュファイル情報記憶手段を含み、

前記キャッシュ管理手段は、前記記憶手段に記憶されたリクエストデータの総容量が所定値を超えた場合に前記総容量が前記所定値以下になるまで前記アクセスされた時刻が古いキャッシュファイルを順次消去する、請求項7記載のネットワークキャッシュ装置。

【請求項9】 前記キャッシュ処理手段はさらに、前記キャッシュファイルの書き込みオープン状態が記憶されるキャッシュファイル情報記憶手段を含み、

前記キャッシュ管理手段は、前記ネットワークインタフェース手段によって受信されたレスポンスデータをキャッシュファイルに記憶する際、前記キャッシュファイル情報記憶手段に記憶された書き込みオープン状態を参照して前記レスポンスデータを記憶するキャッシュファイルを決定する、請求項7記載のネットワークキャッシュ装置。

【請求項10】 ネットワークに対してデータアクセスのリクエストを発行し、該リクエストに対するレスポンスデータを受信するステップと、

前記ネットワークから受信したレスポンスデータを記憶するステップと、

アプリケーションからプロセス間通信によりデータリクエストを受信し、

前記データリクエストに対応するレスポンスデータが記憶されている場合は、前記ネットワークに対してデータアクセスのリクエストを発行せず、

前記データリクエストに対応するレスポンスデータが記憶されていない場合は、前記ネットワークに対してデータアクセスのリクエストを発行するステップとを含むネットワークキャッシュ方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ネットワークを介して外部のデータにアクセスする情報端末装置に関する、特に、ネットワークから受信したデータを内部にキャッシュするネットワークキャッシュ装置、ネットワークキャッシュ方法およびネットワークキャッシュプログラムを記録した媒体に関する。

【0002】

【従来の技術】 近年、ワークステーション、パソコンコンピュータあるいは電子手帳等の情報端末装置が普及し、これらの情報端末装置を利用してネットワーク上のサーバ計算機が保持する情報をアクセスできるネットワークファイルシステムが種々ユーザに提供されている。その代表的なものとして、いわゆるインターネットにおけるH T T P (Hyper Text Transfer Protocol) を使用した広域分散型マルチメディア情報システムがある。

【0003】 インターネットは、その基本プロトコルと

してT C P / I P (Transmission Control Protocol / Internet Protocol) を利用したグローバルなネットワークである。インターネット上に構築された地域分散型マルチメディア情報システムは、W W W (World Wide Web) と呼ばれ、ネットワーク上に分散したファイルオブジェクトを扱うことができる。これらのファイルオブジェクトは、単なるテキストデータに留まらず、画像データ、音声データ、ビデオ画像データなどさまざまな種類のものを含む。W W W は、情報提供側にとってもユーザ側にとっても魅力的であるため、ネットワーク上のW W W に関するトラフィックが爆発的に増加している。W W W システムでは、クライアント計算機のユーザは、グラフィカルユーザインタフェースを持ったブラウザソフトウェアをクライアント計算機上で実行させるだけで、ネットワーク上に分散したサーバ計算機の保持するさまざまなファイルオブジェクトで構成された情報を次々とアクセスすることができる。W W W は、このような操作の簡単さのために近年目ざましい普及を見せている。

【0004】 ユーザは、W W W ブラウザを使用することにより、ネットワークからデータを受信して画面に表示することが可能となる。W W W ブラウザにはユーザの操作性の向上のためにさまざまな機能が組込まれている。

【0005】 一般的に使用されているW W W ブラウザの1つには、ネットワークから受信したデータをキャッシュするネットワークキャッシュ機能が組込まれている。ユーザは、このネットワークキャッシュ機能を利用するにより、一度ネットワークから受信データデータをアクセスする場合にキャッシュされたデータがアクセスされてアクセス時間を短縮させることができる。

【0006】 また、特開平4-219839号公報に開示された発明は、ネットワークキャッシュ装置および計算機システムを含むネットワークキャッシュシステムに関するものであり、その概略構成を図18に示す。ネットワークキャッシュ装置70は、ネットワークインタフェース71と、ネットワークインタフェース71を介して受信したパケットの処理およびネットワークインタフェース71を介してパケットの送信を行なうためのパケット処理部72と、ファイル格納エリア74と、ファイル格納エリア74に格納されているファイルに関する情報を登録するファイル情報格納エリア73と、ファイルバッファ75とを含む。

【0007】 また、計算機システム80または90は、ネットワークインタフェース83または93と、アクセスされるファイルがどこにあるかを調査するためのファイルアクセス処理部81または91と、ネットワークインタフェース83または93を介して受信したパケットの処理およびネットワークインタフェース83または93を介してパケットの送信を行なうためのパケット処理部82または92と、ローカルディスク上にあるファイルでかつネットワークキャッシュ装置70のファイル格

納エリア74にも格納されていて頻繁にアクセスされるファイルの名前を登録するためのファイルキャッシュテーブル84または94と、ローカルディスク上にあるファイルに対するアクセスの処理を行なうためのローカルファイルアクセス処理部85または95とを含む。

【0008】頻繁にアクセスされるファイルの名前が計算機システム80または90のファイルキャッシュテーブル84または94に登録される。計算機システム80のファイルアクセス処理部81は、要求されるファイルの格納場所を調査し、ファイルキャッシュテーブル84に登録されていることを検出すれば、パケット処理部82がネットワークキャッシュ装置70に対してファイルアクセス要求パケットを送出する。そしてネットワークキャッシュ装置70は、ファイル格納エリア74に格納されているファイルをパケットにコピーしてアクセス要求元の計算機80に送出する。

【0009】

【発明が解決しようとする課題】上述したWWWブラウザの1つでは、ネットワークキャッシュ機能がブラウザ内に組込まれているので、2つのブラウザが同時にキャッシュデータを共有したり、他のアプリケーションがキャッシュされたデータを用いるのは困難であるという問題点がある。

【0010】また、キャッシュデータは1データにつき1ファイルという形式で格納されており、電子手帳などの携帯端末などで使用できるファイル数に制限があるシステムの場合には、このままでは組込めないという問題点がある。

【0011】また、上述した特開平4-219839号公報に開示された発明においては、キャッシュされるデータはファイルアクセス要求を発行する計算機システム80または90内ではなく、ネットワークキャッシュ装置70内に格納されるため、キャッシュされたデータをアクセスするためには計算機システム80または90とネットワークキャッシュ装置70間のネットワークを介した通信が必要であり、計算機システム80または90がデータを得るのに時間がかかるという問題点がある。

【0012】本発明は、上記問題点を解決するためのなされたもので、請求項1～9に記載の発明の目的は、複数のアプリケーションが同じキャッシュデータを共有可能であり、各アプリケーションがレスポンスデータを速く受信することが可能なネットワークキャッシュ装置を提供することである。

【0013】請求項10に記載の発明の目的は、複数のアプリケーションが同じキャッシュデータを共有可能であり、各アプリケーションがレスポンスデータを速く受信することが可能なネットワークキャッシュ方法を提供することである。

【0014】請求項11に記載の発明の目的は、複数のアプリケーションが同じキャッシュデータを共有可能で

あり、各アプリケーションがレスポンスデータを速く受信することが可能なネットワークキャッシュプログラムを記録した媒体を提供することである。

【0015】

【課題を解決するための手段】請求項1に記載のネットワークキャッシュ装置は、ネットワークに対してデータアクセスのリクエストを発行し、リクエストに対するレスポンスデータを受信するためのネットワークインタフェース手段と、ネットワークインタフェース手段が受信したレスポンスデータを記憶するための記憶手段と、アプリケーションからプロセス間通信によりデータリクエストを受信し、データリクエストに対応するレスポンスデータが記憶手段に記憶されている場合は、ネットワークインタフェース手段に対してデータアクセスのリクエストの発行を指示せず、データリクエストに対応するレスポンスデータが記憶手段に記憶されていない場合は、ネットワークインタフェース手段に対してデータアクセスのリクエストの発行を指示するためのキャッシュ処理手段とを含む。

【0016】キャッシュ処理手段は、アプリケーションからプロセス間通信により受信したデータリクエストに対応するレスポンスデータが記憶手段に記憶されている場合は、ネットワークインタフェース手段に対してデータアクセスのリクエストを発行しないので、アプリケーションがレスポンスデータを速く受信することができる。

【0017】請求項2に記載のネットワークキャッシュ装置は、請求項1に記載のネットワークキャッシュ装置であって、キャッシュ処理手段は、リクエストに対応するリクエストデータ名とリクエストに対応するレスポンスデータが記憶手段内に記憶される領域とがインデックス情報として記憶されるインデックス記憶手段と、ネットワークインタフェース手段によって受信されたレスポンスデータを記憶手段に記憶し、記憶されたレスポンスデータに対応するインデックス情報をインデックス記憶手段に記憶するためのキャッシュ管理手段と、アプリケーションからプロセス間通信によりデータリクエストを受信し、データリクエストに対応するリクエストデータ名がインデックス記憶手段にある場合は、記憶手段に記憶されたデータリクエストに対応するリクエストデータをプロセス間通信によりアプリケーションへ送出し、データリクエストに対応するリクエストデータ名がインデックス記憶手段にない場合は、ネットワークインタフェース手段に対してデータアクセスのリクエストの発行を指示するためのアプリケーションリクエスト処理手段と、ネットワークインタフェース手段によって受信されたレスポンスデータをプロセス間通信によりアプリケーションへ送出するためのレスポンスデータ処理手段とを含む。

【0018】アプリケーションリクエスト処理手段は、

アプリケーションからプロセス間通信により受信したデータリクエストに対応するリクエストデータ名がインデックス記憶手段にある場合は、記憶手段に記憶されたデータリクエストに対応するリクエストデータをプロセス間通信によりアプリケーションへ送出するので、アプリケーションがレスポンスデータを速く受信することが可能となる。

【0019】請求項3に記載のネットワークキャッシュ装置は、請求項2に記載のネットワークキャッシュ装置であって、レスポンスデータ処理手段は、ネットワークインターフェース手段によって受信されたレスポンスデータを共有メモリ領域にコピーし、レスポンスデータがコピーされた領域へのポインタとレスポンスデータのデータサイズとをアプリケーションへメッセージ通信する。

【0020】レスポンスデータ処理手段は、アプリケーションに対してレスポンスデータがコピーされた領域へのポインタとレスポンスデータのデータサイズのみをメッセージ通信するので、アプリケーションがレスponsデータへアクセスする速度が速くなる。

【0021】請求項4に記載のネットワークキャッシュ装置は、請求項2に記載のネットワークキャッシュ装置であって、キャッシュ処理手段はさらに、ネットワークインターフェース手段がネットワークに対してデータアクセス中のリクエストの情報を記憶されるネットワークアクセス情報記憶手段を含み、アプリケーションリクエスト処理手段はアプリケーションからプロセス間通信によりデータリクエストを受信し、データリクエストに対応するリクエストデータ名がインデックス記憶手段にある場合は、記憶手段に記憶されたデータリクエストに対応するリクエストデータをプロセス間通信によりアプリケーションへ送出し、データリクエストに対応するリクエストデータ名がインデックス記憶手段になく、ネットワークアクセス情報記憶手段にもない場合は、ネットワークインターフェース手段に対してデータアクセスのリクエストの発行を指示し、ネットワークアクセス情報記憶手段に当該データリクエストの情報を記憶し、データリクエストに対応するリクエストデータ名がインデックス記憶手段になく、ネットワークアクセス情報記憶手段にある場合は、ネットワークアクセス情報記憶手段に記憶されている当該リクエストのデータアクセスが終了するまでデータリクエストに対する処理を延期する。

【0022】アプリケーションリクエスト処理手段は、アプリケーションからプロセス間通信により受信したデータリクエストに対応するリクエストデータ名がインデックス記憶手段になく、ネットワークアクセス情報記憶手段にある場合は、ネットワークアクセス情報記憶手段に記憶されている当該リクエストのデータアクセスが終了するまでデータリクエストに対する処理を延期する。したがって、アプリケーションは当該リクエストのデータアクセスが終了すれば、このデータアクセスによって

得られたリクエストデータを受信することが可能になる。

【0023】請求項5に記載のネットワークキャッシュ装置は、請求項2に記載のネットワークキャッシュ装置であって、キャッシュ管理手段は、インデックス記憶手段にインデックス情報を記憶する際記憶手段にも同じインデックス情報を記憶し、記憶手段に格納されたリクエストデータを更新する際更新された情報を付加したインデックス情報を記憶手段に追加し、所定のタイミングで更新された情報を付加したインデックス情報を参照して記憶手段に記憶されたインデックス情報を再構築する。

【0024】キャッシュ管理手段は、リクエストデータを更新する際更新された情報を付加したインデックス情報を記憶手段に追加するのみであるので、処理速度が速くなり、アプリケーションがレスポンスデータを速く受信することが可能となる。

【0025】請求項6に記載のネットワークキャッシュ装置は、請求項2に記載のネットワークキャッシュ装置であって、キャッシュ管理手段は、インデックス記憶手段にインデックス情報を記憶する際当該リクエストデータがアクセスされた時刻も記憶し、記憶手段に記憶されたリクエストデータの総容量が所定値を超えた場合に、総容量が所定値以下になるまでアクセスされた時刻が古いリクエストデータを順次消去する。

【0026】キャッシュ管理手段は、記憶手段に記憶されたリクエストデータのうち時刻が古いリクエストデータを順次消去するので、効率よくリクエストデータをキャッシュすることが可能となる。

【0027】請求項7に記載のネットワークキャッシュ装置は、請求項2に記載のネットワークキャッシュ装置であって、キャッシュ管理手段は、ネットワークインターフェース手段によって受信されたレスポンスデータを記憶手段に記憶する際、1つのキャッシュファイルに複数のレスポンスデータを記憶する。

【0028】キャッシュ管理手段は、1つのキャッシュファイルに複数のレスポンスデータを記憶するので、キャッシュファイル数を減らすことが可能となる。したがって、キャッシュファイル数に制限があるシステムの場合でも、効率よくレスポンスデータをキャッシュすることが可能となる。

【0029】請求項8に記載のネットワークキャッシュ装置は、請求項7に記載のネットワークキャッシュ装置であって、キャッシュ処理手段はさらに、キャッシュファイルがアクセスされた時刻がキャッシュ管理手段によって記憶されるキャッシュファイル情報記憶手段を含み、キャッシュ管理手段は、記憶手段に記憶されたリクエストデータの総容量が所定値を超えた場合に総容量が所定値以下になるまでアクセスされた時刻が古いキャッシュファイルを順次消去する。

【0030】キャッシュ管理手段は、アクセスされた時

刻が古いキャッシュファイルを順次消去するので、効率よくキャッシュファイルを作成することが可能となる。

【0031】請求項9に記載のネットワークキャッシュ装置は、請求項7に記載のネットワークキャッシュ装置であって、キャッシュ処理手段はさらに、キャッシュファイルの書込オープン状態が記憶されるキャッシュファイル情報記憶手段を含み、キャッシュ管理手段は、ネットワークインターフェース手段によって受信されたレスポンスデータをキャッシュファイルに記憶する際、キャッシュファイル情報記憶手段に記憶された書込オープン状態を参照してレスポンスデータを記憶するキャッシュファイルを決定する。

【0032】キャッシュ管理手段は、キャッシュファイル情報記憶手段に記憶された書込オープン状態を参照してレスポンスデータを記憶するキャッシュファイルを決定するので、キャッシュファイルの内容が破壊されるのを防ぐことが可能となる。

【0033】請求項10に記載のネットワークキャッシュ方法は、ネットワークに対してデータアクセスのリクエストを発行し、リクエストに対するレスポンスデータを受信するステップと、ネットワークから受信したレスポンスデータを記憶するステップと、アプリケーションからプロセス間通信によりデータリクエストを受信し、データリクエストに対応するレスポンスデータが記憶されている場合は、ネットワークに対してデータアクセスのリクエストを発行せず、データリクエストに対応するレスポンスデータが記憶されていない場合は、ネットワークに対してデータアクセスのリクエストを発行するステップとを含む。

【0034】

【発明の実施の形態】図1は、本発明の実施の形態におけるネットワークキャッシュ装置の構成を示すブロック図である。ネットワークキャッシュ装置は、ネットワーク8と接続されており、装置全体を制御するためのCPU (Central Processing Unit) 1と、アプリケーションおよびネットワークキャッシュ装置を制御するプログラムを記憶するためのメモリ2と、ユーザがアプリケーションを操作するためのキーボード、マウスあるいはペンなどの入力装置3と、画面表示を行なうためのディスプレイ4と、ハードディスクなどの外部記憶装置5と、ネットワーク8に対してデータアクセスのリクエストを発行し、リクエストに対するレスポンスデータを受信するためのネットワーク装置6と、ネットワークキャッシュ装置を制御するプログラムを記録した媒体7の駆動部(図示せず)とを含む。

【0035】ネットワークキャッシュ制御プログラムは、CPU1によって一端外部記憶装置5に格納される。CPU1は、適宜外部記憶装置5からネットワークキャッシュ制御プログラムをメモリ2内にロードして実行することによってネットワークキャッシュ装置が動作

を行なう。

【0036】メモリ2は、複数のアプリケーション22と、ネットワークキャッシュ処理部(ネットワークキャッシュ制御プログラム)とを含む。さらに、ネットワークキャッシュ処理部21は、ネットワーク装置6に対してデータアクセスのリクエストを発行するよう指示をし、このリクエストに対するレスポンスデータをネットワーク装置6から受取るためのネットワークインターフェース部25、ネットワークインターフェース部25からレスポンスデータを受取ってキャッシュデータとして外部記憶装置5に格納し、外部記憶装置5内に格納された当該キャッシュデータの格納場所等を管理するためのキャッシュ管理部26、アプリケーション22からのデータアクセスのリクエストを受信し必要に応じてネットワークインターフェース部25に対してデータアクセスのリクエストの発行を指示するためのアプリケーションリクエスト処理部27、キャッシュ管理部26からレスポンスデータを受取りアプリケーション22へ送出するためのレスポンスデータ処理部28、外部記憶装置5内にキャッシュされたデータのインデックス情報を記憶するためのキャッシュインデックス記憶部201、外部記憶装置5内のキャッシュファイルの名前、サイズ等の情報を記憶するためのキャッシュファイル情報記憶部202、およびネットワーク8に対してアクセス中のデータの情報を記憶するためのネットワークアクセス情報記憶部203を含む。

【0037】また、外部記憶装置5は、キャッシュデータを記憶するためのキャッシュデータ記憶領域51と、キャッシュデータのインデックス情報を記憶するためのキャッシュインデックス記憶領域52とを含む。

【0038】図2は、キャッシュデータ記憶領域51に記憶されるキャッシュファイルの構造を示している。キャッシュファイルは、ネットワーク8を介して受信した1レスポンスデータにつき1ファイルの構成も可能であるが、本実施の形態では図2に示すように1つのファイルに複数のレスポンスデータをまとめて記録する構成を採る。また、使用されるファイルシステムに合わせてキャッシュファイル数の制限値(以下、CACHE_NUM_LIMITと称す)およびキャッシュファイルの総容量制限値(以下、CACHE_SIZE_LIMITと称す)が予め設定される。

【0039】図3は、キャッシュインデックス記憶部201に記憶されるキャッシュインデックス情報のデータ構造の一例を示している。キャッシュインデックス記憶部201にはキャッシュされているデータのそれぞれについて、キャッシュされているデータの名称(リクエストデータ名)、当該データが格納されているキャッシュファイル名、キャッシュファイルの先頭に対する当該データの先頭オフセット値、キャッシュファイルの先頭に対する当該データの末尾位置のオフセット値、および当

該データが最後にアクセスされた時刻が記憶されている。

【0040】図4は、キャッシュファイル情報記憶部202に記憶されるキャッシュファイル情報のデータ構造の一例を示している。キャッシュファイル情報記憶部202にはキャッシュファイルのそれぞれについて、キャッシュファイル名、キャッシュファイルのファイルサイズ、キャッシュファイルが書き込まれ中であるか否かを示すフラグ、および当該キャッシュファイルが最後にアクセスされた時刻が記憶されている。なお、キャッシュファイルは初期状態では、キャッシュファイル数の制限値(CACHE_NUM_LIMIT)の数だけ予めキャッシュファイル名を決めておき、ファイルサイズ0で生成しておくものとする。

【0041】図5は、ネットワークアクセス情報記憶部203に記憶されるネットワークアクセス情報のデータ構造の一例を示している。ネットワークアクセス情報記憶部203にはその時点でネットワークに対してオープンがかかっているそれぞれのリクエストについて、オープンをかけたアプリケーションID、リクエストID、リクエストされているデータの名称(リクエストデータ名)、リクエストデータが書き込まれるキャッシュファイル名、キャッシュファイルの先頭に対するリクエストデータの先頭のオフセット値、およびネットワークに対してオープンをかけたときのネットワークオープンID(たとえば、通信ソケットのファイルディスクリプタ)が記憶されている。

【0042】図6(a)はアプリケーション22からネットワークキャッシュ処理部21にプロセス間通信キューを利用して送出されるデータリクエストメッセージを、図6(b)はネットワークキャッシュ処理部21からアプリケーション22に送出されるレスポンスマッセージのデータ構造の一例を示す図である。データリクエストメッセージにはリクエストを発行したアプリケーションID(その値はレスポンスを受取るメッセージキーのIDをそのまま用いる)、アプリケーション22が付加するリクエストID、およびリクエストされるデータの名称(リクエストデータ名)が書きかれている。また、レスポンスマッセージには、リクエストID、レスポンスデータが格納される共有メモリ等の先頭アドレスおよびレスポンスデータのサイズが書きかれている。

【0043】図7は、ネットワークキャッシュ処理部21の処理手順の概略を示すフローチャートである。ネットワークキャッシュ処理部21は、通常アプリケーションリクエスト処理部27にアプリケーション22からデータリクエストメッセージが届くというイベントと、ネットワークインタフェース部25にネットワークからデータが到着するというイベントとを監視するイベント待ちにある(step00)。イベントが発生すれば、当該イベントがメッセージイベントであるか否かを判定

し、メッセージイベントであれば(step01, yes)、メッセージイベント処理を行ない(step02)、再びイベント待ちとなる(step00)。また、当該イベントがメッセージイベントでなければ(step01, no)、ネットワーク8からのデータ到着イベントであるか否かを判定し当該イベントがデータ到着イベントであれば(step03, yes)、データ到着イベント処理を行ない(step04)、再びイベント待ちとなる(step00)。また、当該イベントがデータ到着イベントでなければ(step03, no)、再びイベント待ちとなる(step00)。

【0044】図8は、図7に示すメッセージイベント処理の処理手順を示すフローチャートである。アプリケーション22がデータアクセスのリクエストを発行するとき、アプリケーション22はプロセス間通信キューを利用してアプリケーションリクエスト処理部27に対して図6(a)に示すようなデータリクエストメッセージを送出する。アプリケーションリクエスト処理部27がデータリクエストメッセージを受取ると、図3に示すキャッシュインデックス情報を検索し、キャッシュインデックス情報のリクエストデータ名にアプリケーション22が要求してきたリクエストデータ名と一致する項目があるか否かを調べる(step11)。

【0045】キャッシュインデックス情報内に一致するリクエストデータ名があれば(step11, yes)、キャッシュデータ記憶領域51内に今回のリクエストデータが存在することを表わしている。この場合、アプリケーションリクエスト処理部27は、その項目の最終アクセス時刻の内容をそのときの時刻に更新する(step12)。また、図4に示すキャッシュファイル情報内のオープンしたキャッシュファイルについても最終アクセス時刻の内容をそのときの時刻に更新する(step13)。

【0046】次に、図3に示すキャッシュインデックス情報を参照し、キャッシュデータ記憶領域51内のその項目があるキャッシュファイルをオープンし、データ先頭オフセットからデータ末尾オフセットまでのデータをアプリケーション22からアクセス可能な共有メモリ領域にコピーする(step14)。そして、リクエストID(データリクエストメッセージに書きかれていたリクエストID)と共有メモリ領域に書き込まれた当該キャッシュデータのデータ先頭アドレスおよびデータサイズとを書込んだレスポンスマッセージを作成し、アプリケーションIDが示すアプリケーションのメッセージキューに書き込む(step15)。そして、データ終了メッセージ(たとえば、データリクエストメッセージに書きかれていたリクエストIDおよびデータサイズとして-1を書き込んだレスポンスマッセージ)をアプリケーションIDが示すアプリケーションのメッセージキューに書き込み(step16)、イベント待ちに戻る(step00)

0)。

【0047】アプリケーション22では、このレスポンスマッセージを受信して、レスポンスマッセージ中のリクエストIDを検出することによって、どのリクエストデータに対するレスポンスマッセージなのかがわかる。アプリケーション22は、データ先頭アドレスとデータサイズとから共有メモリに書込まれたリクエストデータにアクセスすることができる。このように、リクエストデータがキャッシュデータ記憶領域51内にある場合は、ネットワーク8に対するデータアクセスが行なわれないのでアクセス速度が速い。

【0048】また、キャッシュインデックス情報内に一致するリクエストデータ名がなければ(step11, no)、キャッシュデータ記憶領域51内にリクエストデータがキャッシュされていないことを表わしている。この場合、アプリケーションリクエスト処理部27は、ネットワークオープン処理を行なう(step17)。

【0049】図9は、図8に示すネットワークオープン処理の処理手順を示すフローチャートである。まず、アプリケーションリクエスト処理部27は、図5に示すネットワークアクセス情報を検索し、ネットワークアクセス情報内にアプリケーション22が要求してきたリクエストデータ名と一致する項目があるか否かを調べる(step21)。

【0050】ネットワークアクセス情報内に一致するリクエストデータ名があれば(step21, yes)、以前のリクエストによって同じリクエストデータをネットワークに対してオープン中(アクセス中)であることを表わしている。このとき、アプリケーションリクエスト処理部27は、図5に示すネットワークアクセス情報をアプリケーションID、リクエストID、リクエストデータおよび書込キャッシュファイル名=SUSPENDというデータを追加する(step22)。このデータは、当該リクエストに対する処理を保留したことを表わしている。この保留については後述する。アプリケーションリクエスト処理部27は、ネットワークインタフェース部25に対して当該リクエストを発行せずに、イベント待ちに戻る(step00)。

【0051】また、ネットワークアクセス情報内に一致するリクエストデータ名がなければ(step21, no)、リクエストデータをネットワークに対してオープンするという一連の処理を行なう。また、同時に平行してネットワークに対してオープンできるリクエストの最大数(以下、NET_OPEN_MAXと称す)を予めキャッシュファイル数の制限値CACHE_NUM_LIMIT以下に設定しておくものとする。

【0052】まず、アプリケーションリクエスト処理部27は、ネットワークアクセス情報を参照し、ネットワークオープン中のリクエスト数をカウントしてその数がNET_OPEN_MAXと等しいか否かを調べる(s

tep23)。ネットワークオープン中のリクエスト数がNET_OPEN_MAXと等しい場合(step23, yes)は、それ以上ネットワークオープンをせずに、エラーレスponsメッセージ(たとえば、データリクエストメッセージに書かれていたリクエストIDおよびデータサイズに0を書込んだレスponsメッセージ)をアプリケーションIDに示されるアプリケーションのメッセージキューに書込む(step24)。

【0053】また、ネットワークオープン中のリクエスト数がNET_OPEN_MAXより小さい場合(step23, no)、アプリケーションリクエスト処理部27は、ネットワーク8からリクエストに対するレスponsデータを受信したときにどのキャッシュファイルにレスponsデータを書込むかを決定する(step25)。具体的には、図4に示すキャッシュファイル情報を探査し、以下の条件を満たすキャッシュファイルcacheXを検出する。

【0054】(1) ファイルサイズが“キャッシュファイルの総容量制限値(CACHE_SIZE_LIMIT)/キャッシュファイル数の制限値(CACHE_NUM_LIMIT)”より小さい。

【0055】(2) 書込オープン中でない。

(3) 上記(1)および(2)の条件を満たすキャッシュファイルの中で最終アクセス時刻が最も最近である。

【0056】なお、NET_OPEN_MAX≤CACHE_NUM_LIMITとしているので、上記(1)～(3)の条件を満たすようなキャッシュファイル項目がキャッシュファイル情報内に必ず存在する。そして、アプリケーションリクエスト処理部27は、図5に示すネットワークアクセス情報に“アプリケーションID=データリクエストメッセージ内のアプリケーションID”、“リクエストID=データリクエストメッセージ内のリクエストID”、“リクエストデータ=データリクエストメッセージ内のリクエストデータ”、“書込キャッシュファイル名=cacheX”、“データ先頭オフセット=キャッシュファイル情報内のcacheXのファイルサイズ”という項目を追加する(step26)。次に、ネットワークインタフェース部25がネットワーク装置6を介してリクエストデータをネットワーク8に対してオープンし、ネットワークアクセス情報内のネットオープンIDにネットワークオープンに対応するIDを記録する(step27)。そして、イベント待ちに戻る(step00)。

【0057】図10は、図7のデータ到着イベント処理の処理手順を示すフローチャートである。ネットワークインタフェース部25は、ネットワーク装置6を介してネットワーク8に対してオープンされたリクエストに対するレスponsデータをネットワーク8から受信する。ネットワーク8からのレスponsデータは、いくつかの

データパケットに分けて送られてくる。そして、最後にデータ終了を表わすデータ終了パケットが送られてくる。キャッシュ管理部26がネットワークインタフェース部25から受取ったレスポンスデータに対するネットオープンIDを、図5に示すネットワークアクセス情報内で検索することによって、どのリクエストに対するレスポンスデータなのかを判別できる（ネットワークアクセス情報内には必ずネットオープンIDが一致する項目が1つだけ存在する。この一致する項目をcとする）。

【0058】ネットワーク8から受信したレスポンスデータがデータパケットである場合（step31, yes）、キャッシュ管理部26はネットワークアクセス情報内の項目cの書込キャッシュファイル名に示されるキャッシュファイルに受信したレスポンスデータを書込み（step32）、キャッシュファイル情報内のキャッシュファイル名がcacheXの項目のファイルサイズの値に今回キャッシュファイルXに書込んだデータサイズを加算して更新する（step33）。次に、レスポンスデータ処理部28が受取ったデータを共有メモリにコピーし（step34）、共有メモリに書込んだデータの先頭アドレス、データサイズおよび項目cのリクエストIDから図6に示すレスポンスマッセージを作成する。そして、項目cのアプリケーションIDに示されるアプリケーション22のメッセージキューに書込み（step35）、イベント待ちに戻る（step00）。

【0059】また、ネットワーク8から受信したレスポンスデータがデータ終了パケットである場合（step31, no）、レスポンスデータの受信終了に伴うリクエスト終了処理（step37）を行なう。

【0060】図11は、図10のリクエスト終了処理の処理手順を示すフローチャートである。キャッシュ管理部26は、ネットワークアクセス情報の項目cを参照し、項目cのリクエストデータ、書込キャッシュファイル名、およびデータ先頭オフセットを用いてキャッシュインデックス情報内に今回のリクエストに関するインデックス情報を追加する。追加したインデックス情報のデータ末尾オフセットは、“キャッシュファイルサイズ-1”とし、最終アクセス時刻にそのときの時刻を書込む（step41）。

【0061】また、キャッシュファイル情報についても最終アクセス時刻をそのときの時刻に更新し、書込オープンフラグをOFFにする（step42）。

【0062】次に、キャッシュ管理部26は、キャッシュファイル情報を参照し、その時点でのキャッシュファイル総容量を計算する（step43）。キャッシュファイル総容量は、キャッシュファイル情報のすべての項目のファイルサイズの総和で求められる。そして、キャッシュファイル総容量がキャッシュファイル総容量制限値CACHE_SIZE_LIMITを超えているか否かを判定する（step44）。キャッシュファイル総

容量がCACHE_SIZE_LIMITを超えている場合（step44, yes）、今回のリクエストデータのキャッシュファイルcacheXへの書込で、キャッシュファイル総容量制限値を超えたことを意味している。この場合、総容量制限値以下になるまで古いキャッシュファイルの消去処理を行なう（step45）。たとえば、キャッシュファイル情報内で、“ファイルサイズが0でない”かつ“書込オープン中でない”ファイルに対して、最終アクセス時刻が古いものから順番に、キャッシュ総容量がキャッシュ総容量制限値CACHE_SIZE_LIMIT以下になるまでキャッシュファイルを消去する。

【0063】ステップstep45のキャッシュファイル消去処理は、キャッシュ管理部26によって行なわれ、キャッシュファイルの中身を消去してキャッシュファイル情報のファイルサイズを0にし、最終アクセス時刻を“00.00.00 00:00:00”に更新する。併せて、キャッシュインデックス情報において、各項目のキャッシュファイル名を調べ、中身を消去したキャッシュファイル内に記憶されていたリクエストデータに関する項目をキャッシュインデックス情報から削除する。

【0064】次に項目cのアプリケーションIDに示されるアプリケーションに対してもデータ終了のレスポンスマッセージをメッセージキューに書込み（step46）、保留リクエスト再開処理を行なう（step47）。保留リクエスト再開処理が終了すると、ネットワークアクセス情報の項目cを消去し（step48）、イベント待ちに戻る（step00）。

【0065】図12は、図11に示す保留リクエスト再開処理の処理手順を示すフローチャートである。ネットワークアクセス情報内に“項目cとリクエストデータ名が同じ”かつ“キャッシュファイル名=SUSPEND”という項目を検索する（step51）。該当する項目がなければ（step51, no）、そのままstep48へ進む。該当する項目があれば（step51, yes）、その項目を項目sとし、項目sのアプリケーションIDとリクエストIDとリクエストデータ名とによってデータリクエストメッセージを作成する（step52）。作成されたデータリクエストメッセージをアプリケーションリクエスト処理部27のデータリクエストメッセージ受信キューに書込み（step53）、ネットワークアクセス情報から項目sを削除する（step54）。そして、ネットワークアクセス情報内に“項目cとリクエストデータ名が同じ”かつ“キャッシュファイル名=SUSPEND”という項目がなくなるまでstep51～step54の処理を繰返す。

【0066】ステップstep53において書込まれたデータリクエストメッセージによって、保留されていたリクエストの処理がアプリケーションリクエスト処理部

21によって再開される。このように、リクエストデータをネットワーク8に対して書込オープンする際に、同じリクエストをネットワーク8に対してオープン中である場合には、後のリクエストに対する処理を先に発行されたりクエストに対する処理が終了するまで延期した後再開することで、後のリクエストはネットワーク8経由でなくキャッシュデータ記憶領域51経由でリクエストデータが返されることで、後のリクエストに対するデータアクセスを速くすることが可能となる。

【0067】本実施の形態においては、図1に示すようにメモリ2内のキャッシュインデックス記憶部201のキャッシュインデックス情報と同じ情報をキャッシュインデックス記憶領域52内にファイルとして記憶される。これは、本実施の形態におけるネットワークキャッシュ装置を再起動したときに、前回までのキャッシュインデックス情報をキャッシュインデックス記憶領域52のキャッシュインデックスファイルから読込むことによって、前回の処理終了時までのキャッシュデータを利用することができる。キャッシュインデックスファイルの更新は、キャッシュインデックス記憶部201内のキャッシュインデックス情報の更新と同時に実行される。概念的には、以下の処理によりキャッシュインデックス情報の更新が行なわれる。

【0068】(1) step37におけるリクエスト終了処理でのキャッシュインデックス情報への項目追加処理(step41)時に、追加した項目をキャッシュインデックス記憶領域52内のキャッシュインデックスファイルにも追加する。

【0069】(2) step37におけるリクエスト終了処理でのキャッシュインデックス情報からの項目削除処理(step45)時に、削除した項目をキャッシュインデックス記憶領域52内のキャッシュインデックスファイルからも削除する。

【0070】(3) step01におけるメッセージイベント処理でのキャッシュインデックス情報の最終アクセス時刻更新処理(step13)時に、キャッシュインデックス記憶領域52内のキャッシュインデックスファイルの最終アクセス時刻も更新する。

【0071】ここで、キャッシュインデックスファイルにおける(1)の追加処理は、キャッシュインデックスファイルに項目情報をアpendすればよい。しかし、(2)の一部削除処理と、(3)の一部更新処理は、ファイルオペレーションでは簡単に実現できない。

【0072】そこで、(2)および(3)の場合に、キャッシュインデックス記憶部21のキャッシュインデックス情報とキャッシュインデックス記憶領域52のキャッシュインデックスファイルの等価性を保つ方法として、一度キャッシュインデックスファイルを消去し、改めてキャッシュインデックス記憶部21のキャッシュインデックス情報をすべてキャッシュインデックスファイル

ルに書き写す(インデックスダンプ)ことが考えられる。

【0073】しかし、(2)および(3)の場合に毎回インデックスダンプを行なうのは、ファイルオペレーションに非常に時間がかかり、アプリケーションへのレスポンスデータの送出が遅くなる。そこで(2)、(3)の場合は、図13に示すようなリクエストデータの変更を示す項目をキャッシュインデックスファイルに追加することにする。図13(a)はリクエストデータの削除を示す項目で、キャッシュファイル名に“DELETE D”が書かれていることで、リクエストデータ“http://www. foo. com”がキャッシュファイルから消去されたことを表わす。

【0074】図13(b)は、リクエストデータの最終アクセス時刻の更新を示す項目で、キャッシュファイル名に“ACCESSED”が書かれていることで、リクエストデータ“http://www. bar. com”が時刻“96.08.3112:15:00”にアクセスされたことを表わしている。

【0075】ただし、その都度項目が追加されて、キャッシュインデックスファイルが肥大化するのを防ぐため、適当なタイミングで(たとえば、(1)、(2)および(3)の処理回数をカウントし、処理回数が所定の数となるごとに、あるいは初期化処理の時点等)インデックスダンプ処理を行なう。

【0076】図14は、キャッシュインデックス情報の初期化処理の処理手順を示すフローチャートである。ネットワークキャッシュ装置の再起動時に前回までのキャッシュデータを利用するためには、キャッシュインデックス情報とキャッシュインデックスファイルの初期化を行ない、前回までのキャッシュインデックス情報を再構築する。

【0077】キャッシュインデックス情報の初期化処理では、キャッシュインデックスファイル内にまだ読出していない項目があるか否かを判定し(step61)、読出していない項目がなくなるまで(step61, no) step62～step67の処理を繰返す。

【0078】まず、キャッシュインデックスファイルから次の1項目を読出す(step62)。読出した項目のキャッシュファイル名に“DELETED”または“ACCESSED”的いずれも書かれていらない場合(step63, yes)は、当該項目は上述したリクエスト終了処理でのキャッシュインデックス情報への項目追加処理時に追加された項目であり、当該項目の内容をそのままキャッシュインデックス情報に登録する(step64)。キャッシュファイル名に“DELETE D”が書かれている場合(step65, yes)は、上述したリクエスト終了処理でのキャッシュインデックス情報からの項目削除処理時に追加された項目であり、当該項目のリクエストデータと同じリクエストデータ名

の項目をキャッシュインデックス情報内で検索し削除する (step 66)。キャッシュファイル名に“ACC E S S E D”が書かれている場合は (step 65, no)、メッセージイベント処理でのキャッシュインデックス情報の最終アクセス時刻更新処理時に追加された項目であり、当該項目の最終アクセス時刻によってその項目のリクエストデータと同じリクエストデータ名を含むキャッシュインデックス情報内の項目の最終アクセス時刻を更新する (step 67)。

【0079】図15は、キャッシュファイル情報の初期化の処理手順を示すフローチャートである。まず、キャッシュファイル情報の一番目からCACHE_NUM_L I M I T番目までのすべての項目について、“ファイルサイズ=当該キャッシュファイルのファイルサイズ”、“書込オープンフラグ=OFF”および“最終アクセス時刻=00. 00. 00 00:00:00”という初期値にセットする (step 71)。キャッシュファイルのファイルサイズは、キャッシュインデックス情報内の同じキャッシュファイル名の項目の中でデータ末尾オフセット値が最大の値を検出し、“1”を加算して求めることができる。

【0080】次に、キャッシュインデックス情報内にまだ読出していない項目があるか否かを判定し (step 72)、読出していない項目がなくなるまで (step 72, no)、step 73～step 75の処理を繰返す。

【0081】まず、キャッシュインデックス記憶部201のキャッシュインデックス情報から次の1項目を読出す (step 73)。読出した項目の最終アクセス時刻 (T1) と読出した項目が格納されるキャッシュファイルのキャッシュファイル情報における最終アクセス時刻 (T2) とを比較する (step 74)。T2よりT1の方が最近である場合 (step 74, yes)、読出した項目が格納されるキャッシュファイルのキャッシュファイル情報における最終アクセス時刻T2をT1に書換える (step 75)。

【0082】図16は、本実施の形態におけるネットワークキャッシュ装置の応用例を示す図である。本応用例は、データ先読みアプリケーション220とWWWブラウザなどのデータブラウザ221とを含む。データ先読みアプリケーション220は、ユーザがこれから見るデータに対してデータブラウザ221とは別にネットワークキャッシュ処理部21に対してデータリクエストを発行する。

【0083】たとえば、データブラウザ221は、予め参照したいリクエストデータのリストを外部記憶装置5内に記憶させておき、ユーザがデータブラウザ221によってデータを参照中に、データ先読みアプリケーション220は同時にリストに挙げられたデータのデータリクエストを順次発行することにより、ネットワークキャ

ッシュ処理部21を介して外部記憶装置5内にリストに挙げられたデータが格納される。このため、ユーザがデータブラウザ221で最初のデータを参照し終え、次のデータを参照する際には図17に示すように、そのデータはネットワーク8からでなく外部記憶装置5から転送される。すなわち、データブラウザ221からのデータリクエストに対しては、ネットワーク8に対するアクセスが発生しないので、ユーザはデータを快適に参照できる。

10 【0084】なお、別アプリケーションでデータ先読みを実現しても、外部記憶装置5に記憶されたキャッシュデータがデータブラウザ221から参照可能なのは、本発明のネットワークキャッシュ装置が複数のプロセスから同時に複数の要求を受付けても、キャッシュデータの整合性を保ちながら動作できるという特徴があるからである。

【0085】なお、本発明ではデータがキャッシュされていれば必ずキャッシュデータが返されるが、ネットワークキャッシュ処理部21において、ネットワーク8を

20 介してまず所望リクエストデータの更新日時を取寄せ、更新されればネットワーク8を介してリクエストデータを受信し、更新されなければキャッシュファイルからリクエストデータを受信するモードを追加することが容易に実現できる。WWWブラウザでよく参照される新聞のページがあるが、ページとしてたとえば天気予報のページやスポーツ結果のページなどよく見るページをリストしておき、上述したモードを用いて先読みされることにより、ユーザはネットワーク8へのアクセス待ちに煩わされることなく所望のページの最新のデータを参照できる。

【0086】また、図16～図17の例では、データ先読みアプリケーション220およびデータブラウザ221の2つのアプリケーションを示しているが、それ以外のアプリケーションも同時にネットワークキャッシュ処理部21にアクセスできる。たとえば上述した2つのアプリケーションと同時に不図示のデータ検索アプリケーションがデータリクエストを発行し、そのデータ内に、あるキーワードが含まれているリクエストデータを探すという処理を行なうことも可能である。

【0087】以上説明したように、本実施の形態におけるネットワークキャッシュ装置によれば次のような効果が得られる。

【0088】アクセス頻度の高いリクエストデータは、キャッシュデータ記憶領域51内にキャッシュデータとして格納され、キャッシュデータ記憶領域51内に記憶されたリクエストデータに対するアクセス要求にかかる通信は、アプリケーション22からのリクエストおよびアプリケーション22へのレスポンスとともにプロセス間通信によって迅速に処理されるので、データアクセス速度を速くすることが可能となる。

【0089】リクエストデータ自体をアプリケーション22へ送出するのではなく、共有メモリ領域へのポインタとデータサイズのみをメッセージ通信でアプリケーション22へ送出するので、データアクセス速度を速くすることが可能となる。

【0090】リクエストデータをネットワーク8に対して書込オープンする際に、同じリクエストをネットワーク8に対してオープン中である場合には、後のリクエストに対する処理を先に発行されたリクエストに対する処理が終了するまで延期した後再開することで、後のリクエストはネットワーク8経由でなくキャッシュデータ記憶領域51経由でリクエストデータが返されることで、後のリクエストに対するデータアクセスを速くすることが可能となる。

【0091】キャッシュインデックス情報をメモリ2および外部記憶装置5上で記憶することによって、電源オフ時やメモリ2内のキャッシュインデックス情報が失われた場合にも、再起動時にファイル形式のキャッシュインデックス情報をメモリ2内のキャッシュインデックス情報に書込むことによってキャッシュインデックス情報を回復でき、前回までのキャッシュデータを利用することができる。また、キャッシュインデックス情報の更新の際に、変更があった内容をキャッシュインデックス情報ファイルに書き加えるのみであるので、毎回キャッシュインデックス情報をファイルに書込むよりもファイルオペレーションが少なくなり、アプリケーション22のデータアクセスを速く処理することが可能となる。

【0092】キャッシュファイルの総容量が制限値を超えてキャッシュデータを消去する際に、最も長い間アクセスされていないリクエストデータから順に消去されるので、アプリケーション22によって頻繁にアクセスされるリクエストデータが優先的にキャッシュデータ記憶領域51に残ることになり、平均のデータアクセスを速くすることが可能となる。

【0093】1つのキャッシュファイルに複数のリクエストデータを書込むことで、使用できるキャッシュファイル数に制限があるシステムにおいてもネットワークキャッシュ装置を構築でき、アプリケーションの平均のデータアクセスを速くすることが可能となる。

【0094】キャッシュファイルの総容量が制限値を超えてキャッシュファイルを消去する際に、最も長い間アクセスされていないキャッシュファイルから順に消去されるので、アプリケーション22によって頻繁にアクセスされるキャッシュファイルが優先的にキャッシュデータ記憶領域51に残されることになり、平均のデータアクセスを速くすることが可能となる。

【0095】並行して複数のデータアクセス要求があった場合には、それぞれのレスポンスデータが同じキャッシュファイルに混ざって保存されることなく、それぞれ別キャッシュファイルに保存されるので、複数のアプリ

ケーション22から並行して複数のデータアクセス要求を受付けることが可能となる。

【図面の簡単な説明】

【図1】本発明の実施の形態におけるネットワークキャッシュ装置の構成を示すブロック図である。

【図2】本実施の形態におけるキャッシュファイルの構成を示す図である。

【図3】キャッシュインデックス記憶部201に記憶されるキャッシュインデックス情報のデータ構造の一例を示す図である。

【図4】キャッシュファイル情報記憶部202に記憶されるキャッシュファイル情報のデータ構造の一例を示す図である。

【図5】ネットワークアクセス情報記憶部203に記憶されるネットワークアクセス情報のデータ構造の一例を示す図である。

【図6】(a)は、アプリケーション22からネットワークキャッシュ処理部21にプロセス間通信を利用して送出されるデータリクエストメッセージのデータ構造の一例を示す図である。(b)は、ネットワークキャッシュ処理部21からアプリケーション22に送出されるレスポンスマッセージのデータ構造の一例を示す図である。

【図7】ネットワークキャッシュ処理部21の処理手順の概略を示すフローチャートである。

【図8】図7に示すメッセージイベント処理の処理手順を示すフローチャートである。

【図9】図8に示すネットワークオープン処理の処理手順を示すフローチャートである。

【図10】図7のデータ到着イベント処理の処理手順を示すフローチャートである。

【図11】図10のリクエスト終了処理の処理手順を示すフローチャートである。

【図12】図11に示す保留リクエスト再開処理の処理手順を示すフローチャートである。

【図13】(a)は、リクエストデータの削除を示す項目を示す図である。(b)は、リクエストデータの最終アクセス時刻の更新を示す項目を示す図である。

【図14】キャッシュインデックス情報の初期化処理の処理手順を示すフローチャートである。

【図15】キャッシュファイル情報の初期化の処理手順を示すフローチャートである。

【図16】本実施の形態におけるネットワークキャッシュ装置の応用例を示す図(その1)である。

【図17】本実施の形態におけるネットワークキャッシュ装置の応用例を示す図(その2)である。

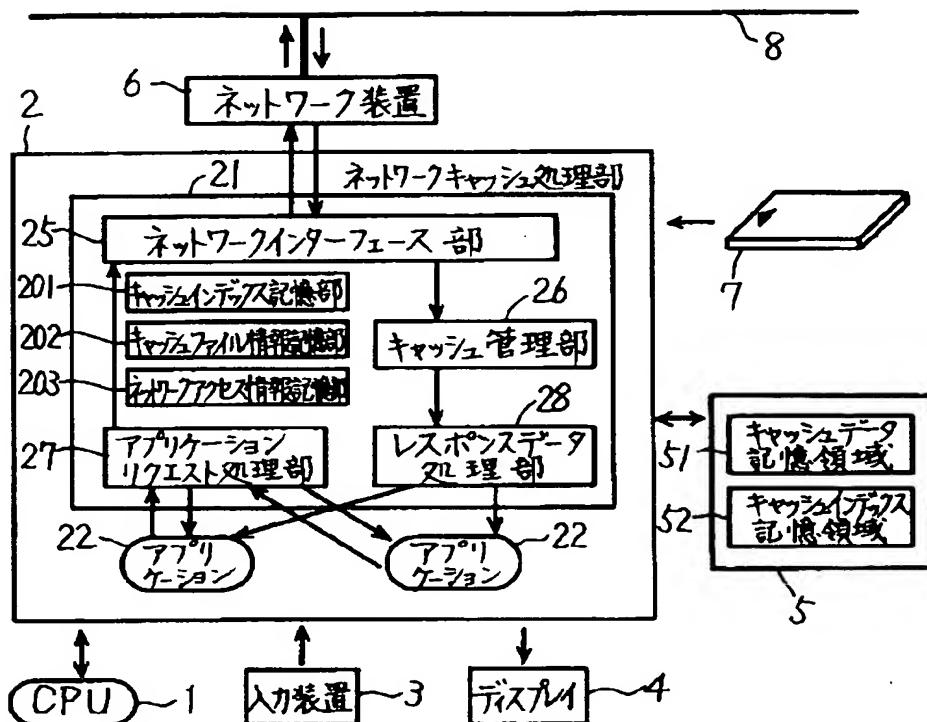
【図18】従来のネットワークキャッシュシステムの概略構成を示す図である。

【符号の説明】

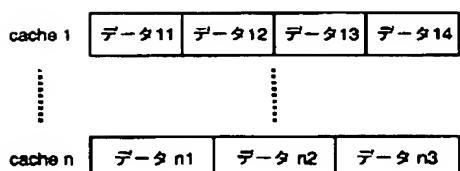
2 メモリ
 3 入力装置
 4 ディスプレイ
 5 外部記憶装置
 6 ネットワーク装置
 7 媒体
 8 ネットワーク
 21 ネットワークキャッシュ処理部
 22 アプリケーション

22 アプリケーション
 25 ネットワークインターフェース部
 26 キャッシュ管理部
 27 アプリケーションリクエスト処理部
 28 レスポンスデータ処理部
 51 キャッシュデータ記憶領域
 52 キャッシュインデックス記憶領域

【図1】



【図2】



【図3】

リクエストデータ	キャッシュファイル名	データ先頭オフセット	データ末尾オフセット	最終アクセス時刻
http://www.foo.com/	cache1	0	1999	96.08.30 12:00:00
http://www.bar.com/	cache1	2000	2053	96.08.30 12:05:30
⋮	⋮	⋮	⋮	⋮

【図4】

キャッシュ ファイル名	ファイル サイズ	書き込みオープン フラグ	最終アク セス時刻
cache1	2054	ON	96.08.30 12:05:30
⋮	⋮	⋮	⋮

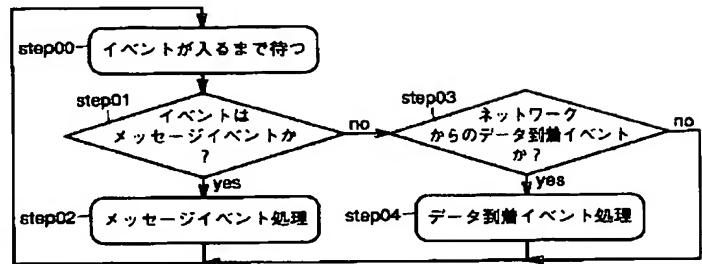
【図5】

アプリケーション ID	リクエストID	リクエストデータ	書き込みキャッシュ ファイル名	データ先頭 オフセット	ネットオープン ID
200	100	http://www.foo.com/	cache1	0	300
⋮	⋮	⋮	⋮	⋮	⋮

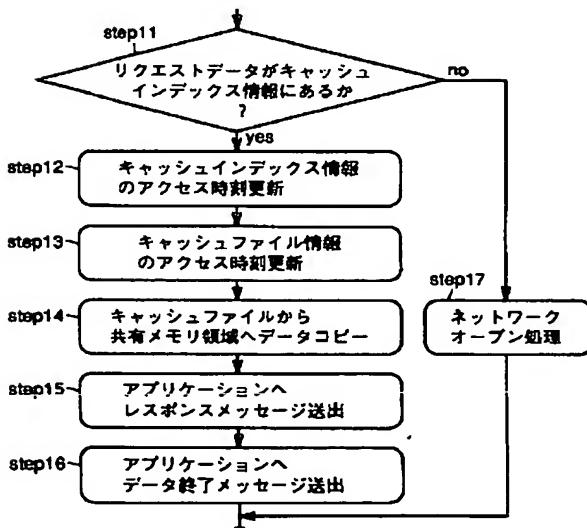
【図6】

(a)	アプリケーションID 200	リクエストID 100	リクエストデータ http://www.foo.com/
データリクエストメッセージ			
(b)	リクエストID 100	データ先頭アドレス 0x1000	データサイズ 2000
レスポンスマッセージ			

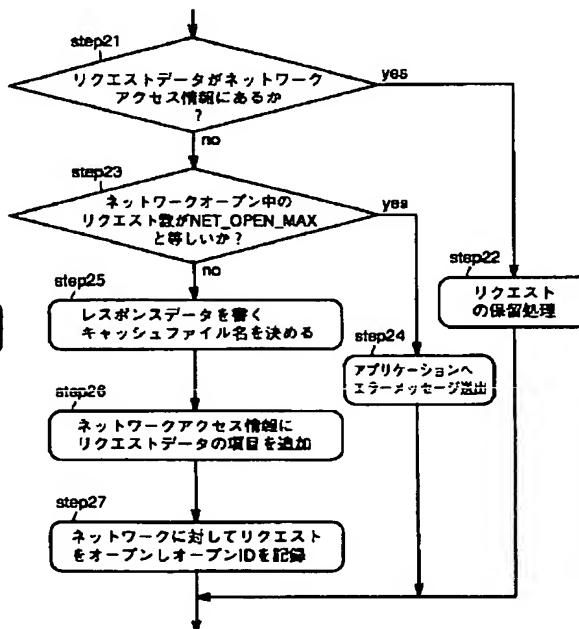
【図7】



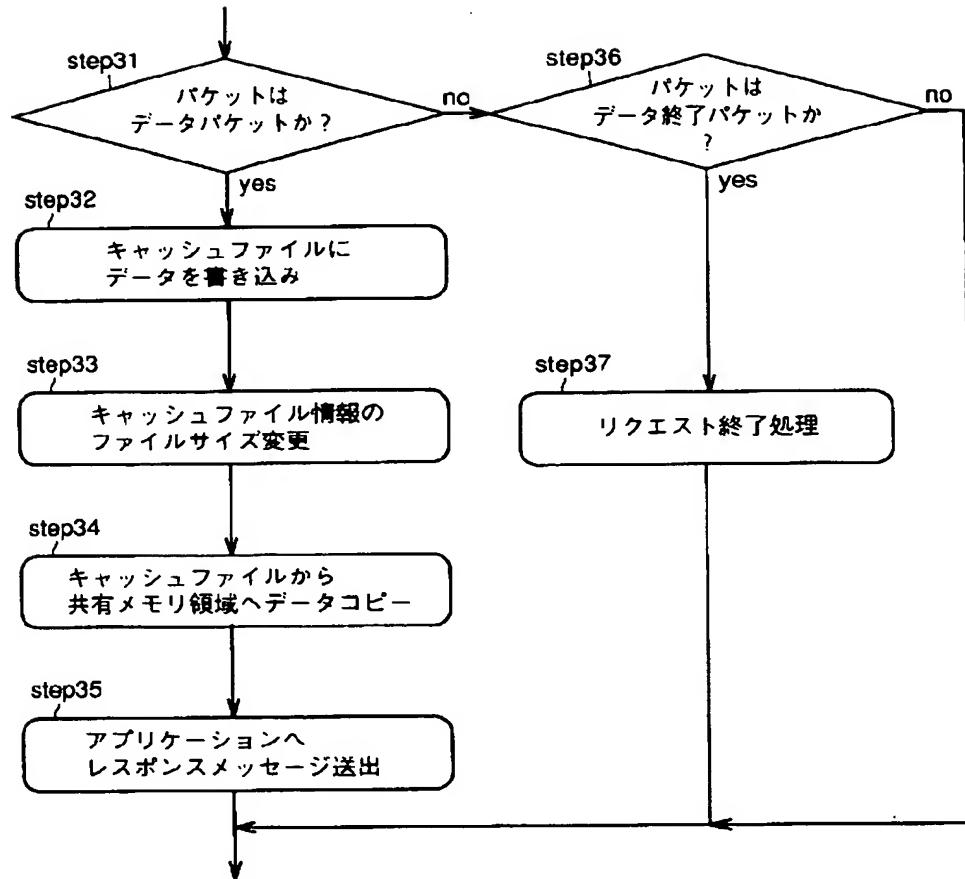
【図8】



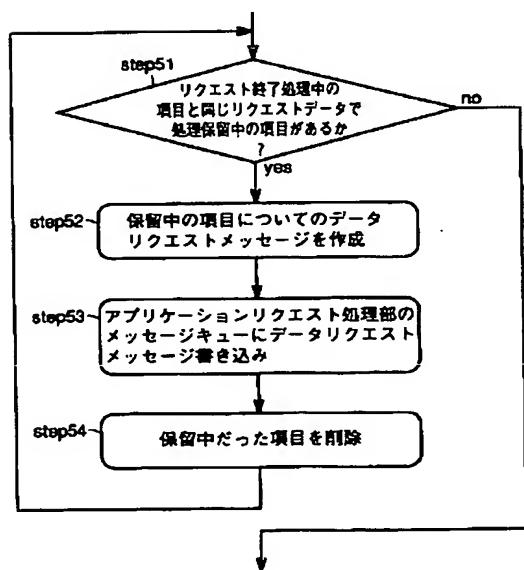
【図9】



【図10】



【図12】



【図13】

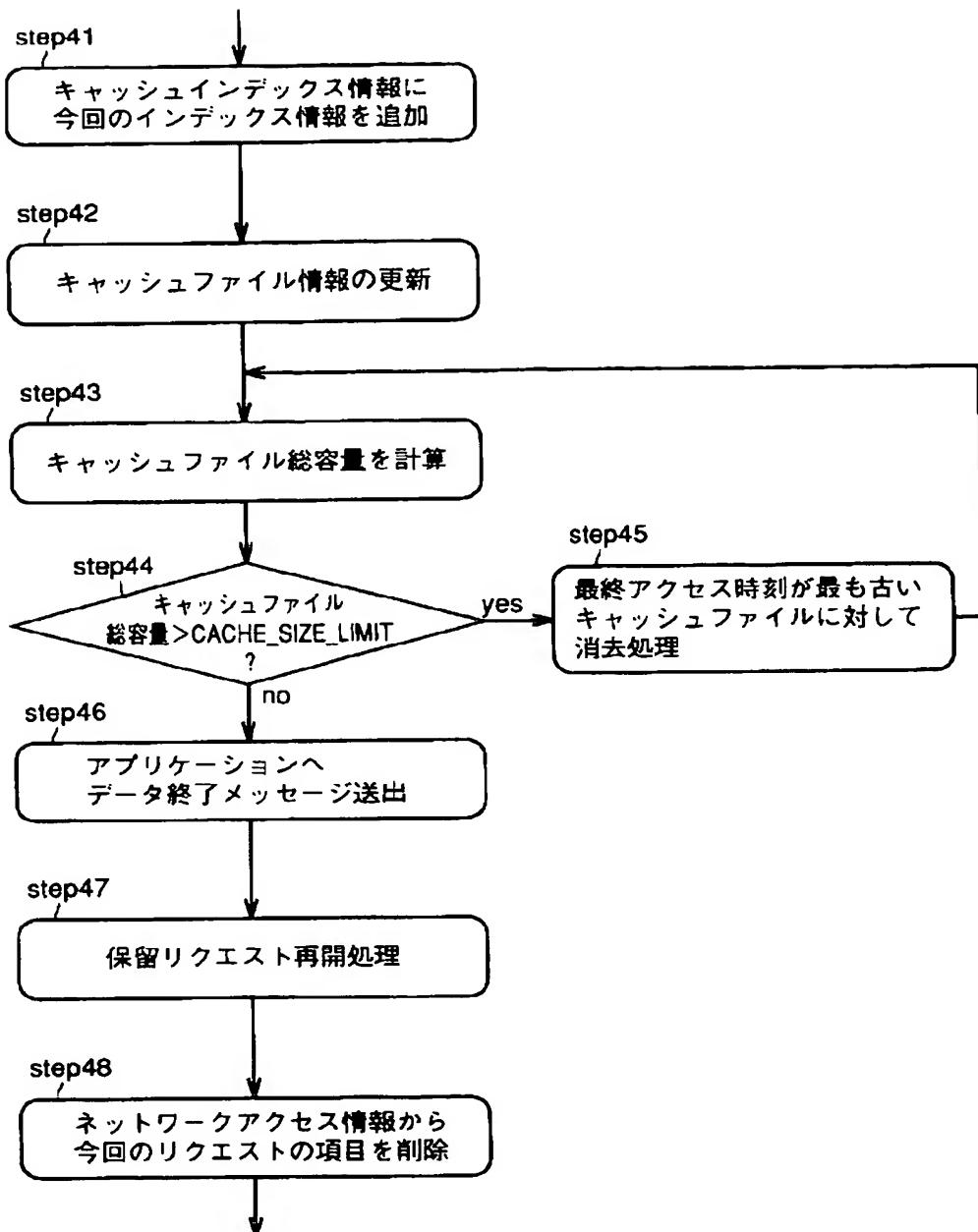
(a)

リクエストデータ	キャッシュ ファイル名	データ先頭 オフセット	データ末尾 オフセット	最終アク セス時刻
http://www.foo.com/	DELETED	0	0	00:00:00 00:00:00

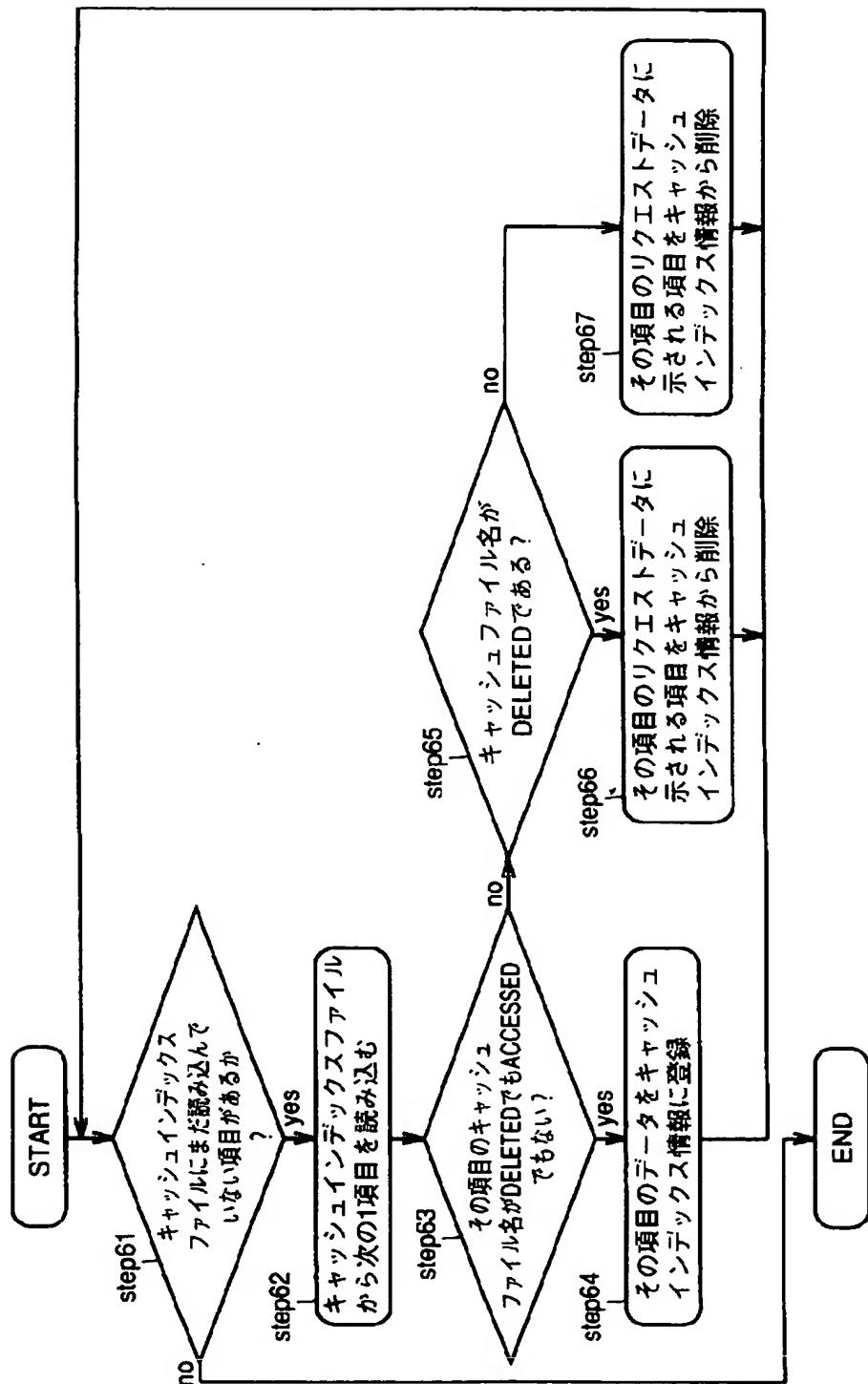
(b)

リクエストデータ	キャッシュ ファイル名	データ先頭 オフセット	データ末尾 オフセット	最終アク セス時刻
http://www.bar.com/	ACCESSD	0	0	98.08.31 12:15:00

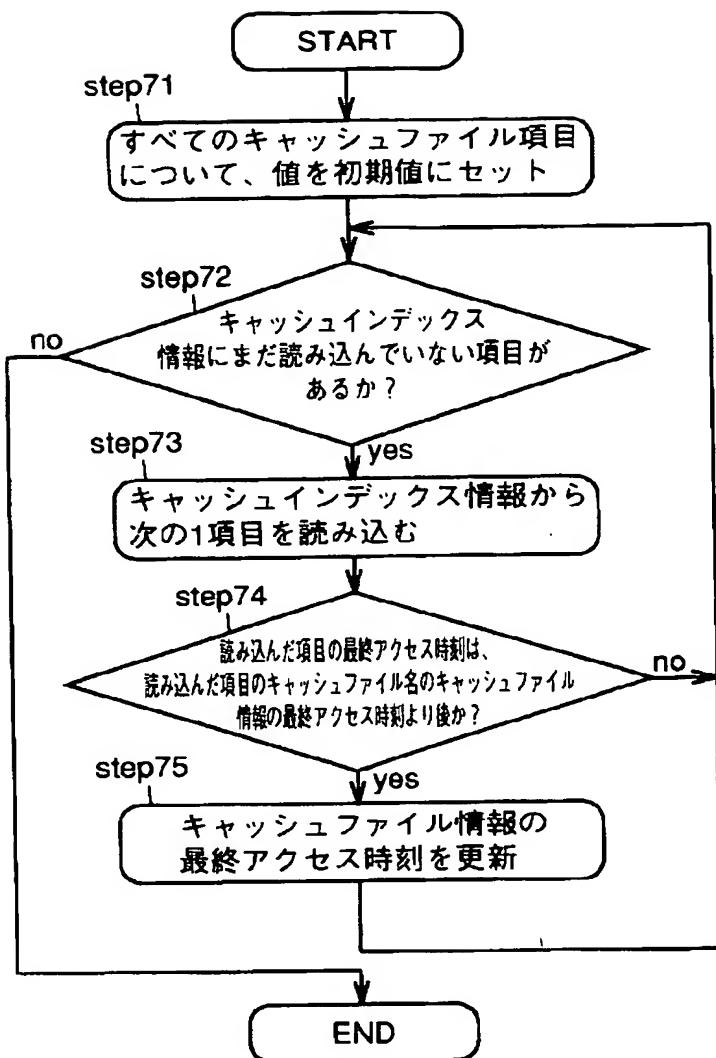
【図11】



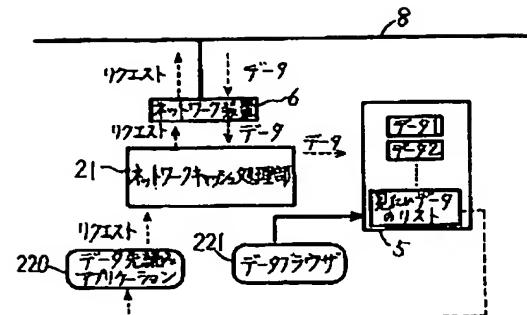
【図14】



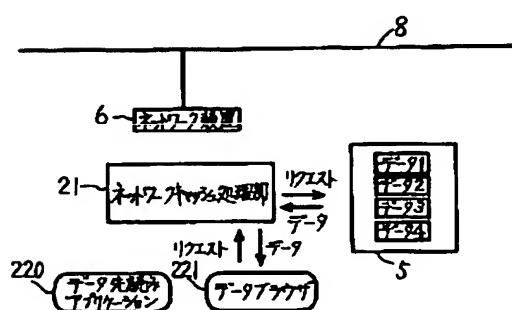
【図15】



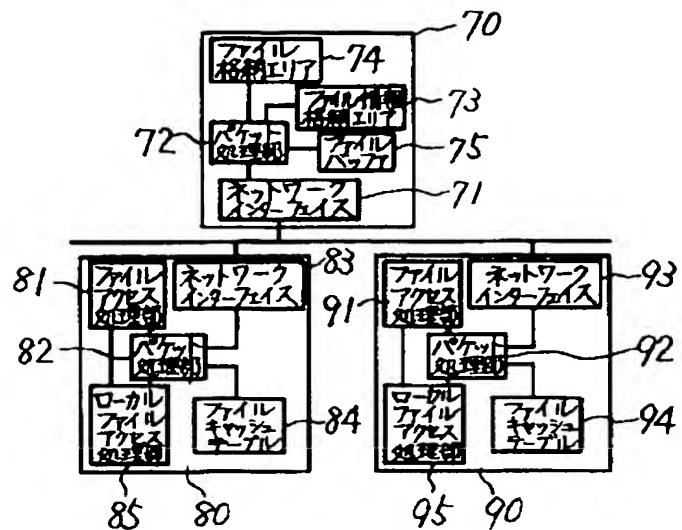
【図16】



【図17】



【図18】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.